Audio Signal Processor

Scholar Team Members:

Cole Aldridge

Lucas Baer

Ron Cummings

Jordan Dahl

Jason Greenwood

Christopher Horne

Tristan Kendall

Greg Middlestead

Brian Rounds

Richard Timm

Aaron Vander Vorst

Faculty Advisor:

Roger Green, Ph.D.

Report for:

August 2004 – May 2005

This document is intended to be read by people outside the group to let them know what the scholar team is doing. It can be read from start to finish, or if the reader so desires they can flip to a section and read only the information pertinent to them.

Mission Statement

This scholar team hopes to successfully design a useful digital signal processing (DSP) board; once this has been accomplished, the group will be presented with myriad opportunities to use the board, seeing as DSPs are extremely versatile. Since versatility was a main goal with a design process involving so many individuals with so many different interests, the board was equipped with a JTAG interface, allowing real-time debugging and limiting the DSP board's use only by the imagination of the engineer. The scholar team hopes to provide a functioning DSP as a spring board for senior design groups in the future, to provide for more powerful processing and versatility in senior design projects and to avoid reliance exclusively on the PIC. Enhanced knowledge of printed circuit board (PCB) design and the use of DSPs will only serve to help the department's students in their knowledge of electrical engineering.

Overall summary

The goal of the DSP scholar team is to successfully design and utilize a digital signal processor mounted on a printed circuit board. The scholar team started by choosing components for the PCB; basing its decision largely on the balance between practical levels of complexity in implementation and processing power, it was decided the Texas Instruments 2812 DSP (TMS30F2812PGF) would be the best option. The team also had to pick out layout software for the PCB. After researching several options it was decided mentor graphics had the best software package based on power, availability, and troubleshooting resources within the department. At this time the team also decided on an appropriate JTAG emulator for the DSP and decided to use code composer studio to work on the software end of the DSP.

The scholar team then began the layout process. The scholar team was able to find a similar board layout, the EzDSP, and decided to base the design for the PCB on the EzDSP. The team had to check and verify the connections for all 176 pins on the DSP to see how they had been laid out for the EzDSP and examine whether any adjustments needed to be made. Once all pin outs were decided, the team was able to start board layout design for the PCB on mentor graphics.

In order to design a printed circuit board layout file on mentor graphics the team had to first make sure all the components for the DSP were available in the parts library on mentor

graphics. The team was provided with a parts library containing basic passive elements such as resistors, capacitors, and inductors of various sizes. However, many components such as the DSP itself, the voltage regulator, etc. were not in the given library, so the team had to design some components on its own; a process which consists of inputting all physical characteristics of the component such as physical width and length, pin layout, and pin size. Once all the necessary parts were at the teams' disposal the team was able to start laying out the board. The team had to specify every connection on the board in mentor graphics. The team also had to decide on the physical layout of the board, tackling such issues as how many layers to use, the geometry and locations of the digital and analog planes, and limitations on physical distance from the JTAG header to the DSP to ensure signal integrity.

This project gave the younger team members a hands-on opportunity to see the design and implementation of a useful electronic device from start to finish. It also afforded some of the younger members a glimpse of what they will be able to do in the future and provided a break from everyday classroom assignments rooted heavily in theory. For the elder members of the group it provided an immense amount of knowledge on the design process. Students were able to learn how to design a PCB from start to finish. Students were exposed to product selection and design issues and students were also able to get first hand experience with a program as powerful as mentor graphics, learning many of the challenges of PCB layout.

The team required some semblance of organization if it was to meet its goal of making and populating a DSP board. Therefore, the 11-member scholar team was split into four subgroups, each focusing on a specific area of complexity on the board design/layout. The four groups are briefly described below, with a little information for each concerning their role in achieving the project goal of a working audio-processing board.

DSP summary

When searching for a Digital Signal Processor the team had to consider the pin count, the amount of on-chip memory, and the company to get it from. The subgroup searched through different company's selection of DSPs and found a list of options. The decision was made to go with Texas Instruments, because of the amount of reference material on-line. The list was narrowed down to the TMS320C28x series. From that point it came down to pin count and memory. Both the 2810 and 2811 models have 128 pins, while the 2812 has 176 pins. However, the 2810 only has 64k ROM, while the 2811 and 2812 have 128k ROM. As a group, it was decided to go with a higher pin count and memory, so that upgrading the board would be easier.

While this added a level of complexity to every level of the design process, the students accepted the challenge with an air of confidence in their ability to ask questions of their advisor.

The DSP subgroup then took on the considerable task of determining connections for each of the 176 pins, as mentioned earlier. While this involved a significant knowledge of circuits and other electrical engineering fundamentals, the less-experienced members of the team (freshmen and sophomores) were able to learn much from their more senior counterparts. Eventually, by working together with the Codec, JTAG, and Layout teams, the pins were all mapped out.

Codec Summary

After choosing which Digital Signal Processor (DSP) that the group was using, and while the DSP team was organizing the pin outs on the DSP, an audio codec had to be chosen and more pins had to be mapped. Compatibility with a 2800 series DSP was an issue that had to be addressed with the codec. Analog to digital as well as digital to analog converting was another key issue which would decide whether or not to accomplish both of those tasks within the same chip or have separate chips for each.

Initially, a 16 bit codec was sought after for many reasons. It was complex enough to do the job of high-quality audio processing, but at the same time was simple enough for under classmen to figure out the workings within. Basing the design off of the EzDSP model, the audio codec from Texas Instruments PCM3003 was chosen. This chip can run in the standard 16 bit mode as well as a 20 bit mode. This was an advantage that it had over the PCM3002, a close contender.

Now that a codec was chosen, compatibility was the next task. Matching up the sampling rate, the bandwidth, the supply voltages, using low pass filters, as well as finding a connection point to the DSP was next. The next task was tracing the leads on the audio daughter card from the previous model to see what exactly someone else had done and what would have to change in this design. Not wanting to use the whole daughter card, there were certain parts of it that needed to be eliminated. Acquiring schematics of the daughter card helped to see what could be eliminated as well as included in this design. Op-amp buffering was needed, but not to the extent that is was on the daughter card. Eliminating jumpers, inverters, and multiple chips to make the design simpler also needed to be done. A new schematic had to be created with leads to and from the DSP as well as the power supply. After all these schematic designs were completed and merged, it was time for the Layout subgroup step up to the plate.

Layout Summary

The layout phase of the project brought together all of the other subgroups in the design and layout of the actual DSP board. The DSP team provided pin-specific connection guidelines which allowed external devices to be combined in the final design. The Codec group provided one of the major external devices, supplying connection specifics and support circuitry schematics for their chip. The JTAG team provided the necessary information which would eventually allow communication to the actual DSP through the JTAG header.

The layout phase began by first giving everyone in the group a tutorial in using Mentor Graphics circuit board design software, as mentioned earlier. Once familiarized with the schematic and PCB layout software, layout members humbly began their design by creating a footprint, or the detailed pattern taken up by a circuit device on a PCB, for each individual component and combining them to their schematic equivalent in the software. Then the real work on the schematic began, as the layout team diligently finalized the schematics on paper and simultaneously began to put them into digital form using Mentor Graphics' schematic software, DesignView.

Three full pages of schematics, lots of LEDs, and many man-hours later, the team took the leap forward into the PCB layout phase, where the senior members shined, and the more junior members of the team soaked up knowledge like a dry sponge in a wet place. Even more man-hours, pages of digitalized thin copper lines, and 45-degree corners later, the layout team felt confident enough of their design to present it to the group for a final check. Before long, the board was ordered, parts were ordered, and the group was populating the PCB.

Note: The final layout design is presented in the appendix, along with a brief users' tutorial for mentor graphics

JTAG summary

The desire was expressed initially to explore real-time debugging through a JTAG connection. To do so would require an emulator, so the purchase of one was pursued. Because an emulator is a large investment (oftentimes ranging into prices greater than $2000), the selection of the appropriate emulator was imperative. After searching and weighing the options, it was decided to maintain a level of certainty and so to purchase what is known to be appropriate for the application. Through connections with Dakota Technology Inc., it was known that they have worked a similar project, so the team decided to purchase the same emulator they use, which is the Spectrum Digital XDS510. The team also made an upgrade to the USB 2.0 model.

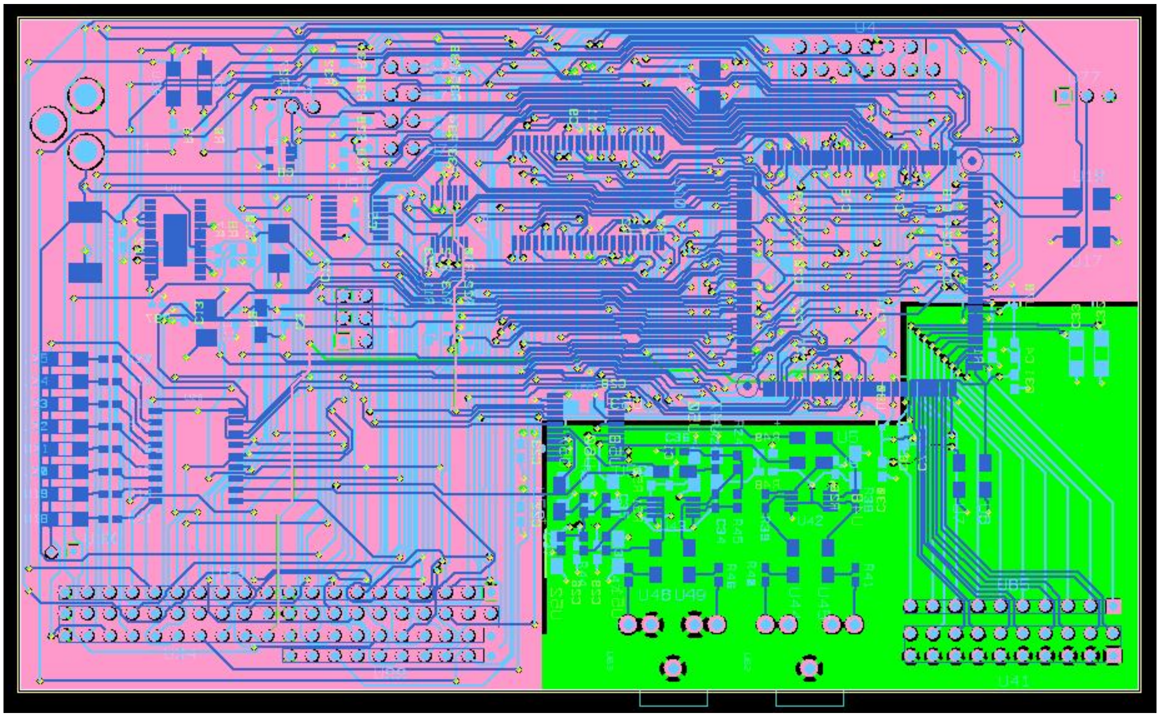
Through contact with Spectrum Digital and examination of schematics of Spectrum Digital DSP boards, the required connections from the DSP to the JTAG header were determined. If the header is placed within two inches of the DSP, then no buffering is required. To keep the project as basic as possible, effort was made on the layout to keep the header and traces less than two inches from the DSP. The header itself was acquired, and is nothing more than a 2x7 generic straight-pin header with one pin clipped.

Focus then shifted to the software aspect. TI's Code Composer Studio (CCS) is the program the team is becoming familiar with to program the DSP. Through CCS tutorials, practice and experimentation was performed on Spectrum Digital's EzDSP board and a parallel port connection to load basic programs onto the DSP. To program the EzDSP (and ultimately this project's board) through an emulator instead of a parallel port required the purchase of a full version of Code Composer Studio, which was then acquired along with a second emulator of the same model. The next goal is to program the DSP to get a hardware response; to make a light blink. The programming will become more sophisticated as familiarity with the process grows to reach the goal of audio processing.

Status Report (as of May 2005)

The next challenge the DSP scholar team intends to take on is the finalization of the board population (trickier than it sounds with the soldering of 0204 traces). The obvious step after that, the one that the team has been working together for all year, is the programming and testing of the populated board. As the DSP scholar team approaches the end of the year, spirits are high and the students are anxiously contemplating the possibilities awaiting them next semester, when implementation will be the name-of-the-game and the fun will really begin.

Appendix:



On this presentation of the layout the digital ground plane is presented in pink and the analog section is green. The virtual separation of ground planes is a fundamental concept in the design of PCBs as it helps insure signal integrity and avoids a plethora of noise issues. It is possible to see where various components are present on the chip, the DSP chip is the easiest to locate, it is the square in the upper right hand corner, notice one portion of the chip is located in the analog plane and the rest is contained in the digital plane.

Mentor Graphics Tutorial

Starting Mentor Graphics:

1. First go to programs in the start menu on the task bar → mentor graphics → S2000 → Library manager for DC-DV expedition → master → got to file and select the library that you will be using → then after selection click on the database icon on the library manager keep library manager open

2. After clicking the database icon, look for the word partition find the brand of the part you will be building → if you find your part, obviously then you do not need to build it → if not present select editor → find the blue and yellow star click it → create new, name part according to what company makes it → Find the name that you just created in the scrollable menu and then select it → again find blue and yellow star, click it → name the part in the number column → name initial → label example: 16/20 bit → find description box in lower right corner → type description in it → keep this window open on other monitor

3. Go back to library manager → click on symbol icon → go to file → new → partition open → magnifying glass → select the partition that you want the part to be saved in or create a new one → go back to the library manager → click on symbol, and type in name → select IC → now make the schematic.

Making the Part Schematic:

1. To make square grab square on the toolbar on the left hand side of the window → to make pins grab the line on the toolbar, it does not like same named pins → connect line to box, make line short → put a pink dot at the end of each pin, the pink dot can be found on the toolbar → press escape to deselect the pink dot tool or any tool → double click on the pink dot at the end of your pin → name the pin → make the pin input, output, or bidirectional depending on the what a footprint recommends and press ok

2. You can click on the red lettering and move it to the inside of the “square”, next to the pin it is associated with (this is common practice)

3. Group pins together that are of the same type (unless the footprint you are using has them laid out neatly already) → ground is usually put on bottom plane → output pins on right, input pins on left (this is common practice)

4. Save as when finished → close this window (one that you made the part in) → find the window that you kept open on the other monitor, the window in step 2 of Starting Mentor Graphics it has the word partition in it → click on pin mapping in the lower right hand corner of this window note: the first pin for some reason has an error → import the partition that you made the part in → select the part that

5. Pin number is important so make sure you number them correctly in pin mapping when finished press ok

6. Go back to library manager → click on cells icon → partition → scroll to the type of IC that yours is for example IC-SSOP#, the number of pins of your part should be same as the number after the P in IC-SSOP# and that is what the # stands for → click on the one that matches your part → the name should be as close a possible to your dimension of your part (go by max), the first number is the width inside, second is length, and third is pad length → input total number of pins and layers → package group → SSOP IC

7. Go to pattern place → pattern type → SOIC → enter dimensions into thousandths (white boxes) → add 15 thousandths to pads for easier soldering → pad stack name → select all pins → when done press place, place it in the middle → make sure measurements are correct → measuring by clicking on the corner once (don’t hold) and move to position where you want to measure, you can tell the distance from where you clicked to where your arrow is currently by looking at the very upper right corner of the window where it should say dy/dx, this is the measurement → save

(not completed)

Making the Layout

1. (Optional) Go back to library manager → click on pack stack icon, this has a lot of stuff to do

2. Go to desktop → click on design view icon → click on project up where file is located → new???????

3. File new → schematic → click on (device) place parts icon, it looks like a chip on the upper tool bar → select burr-brown

4. Click on symbols this is where you find different symbols for the schematic → scroll down to basic and click on it → this is where you will find voltage and ground symbols → click on the one you need

5. Now to connect this voltage or any other symbol to your part → use the green wire tool in the upper tool bar and click from one point of connection to the other, *hint if a lead seems to disappear zoom in or out with the scroll button on the mouse and it will appear if it is still there. Make the schematic as neat as possible

6. Save